

## BAB 2 TELAAH PUSTAKA

Pada bab ini akan dipaparkan mengenai deskripsi *data mining* secara umum dan landasan teori dari algoritma *data mining* yang digunakan pada FIKUI Mining. Selain itu, juga akan dijelaskan mengenai teori mengenai *use case* dan *class diagram*.

### 2.1 Definisi *Data Mining*

Sekumpulan data hanya akan tetap menjadi data yang tidak dapat memberikan informasi apa-apa jika tidak diolah dan diproses dengan benar. Seringkali sekumpulan data hanya tersimpan dengan rapi di sebuah media penyimpanan tanpa dapat memberikan informasi baru yang dapat bermanfaat. Ketika seseorang ingin mengolah data-data yang dia miliki, orang tersebut mungkin juga belum mengetahui pasti bagaimana mengolah data yang dimiliki. Masalah lain yang dihadapi adalah ketika mempunyai sekumpulan data yang cukup besar jumlahnya. Akan membutuhkan waktu dan *resource* yang cukup besar untuk dapat mengolah data tersebut. *Data mining* ini adalah salah satu metode yang dapat digunakan dalam pengolahan dan penganalisisan suatu data.

Secara etimologis, *data mining* terdiri dari dua kata yaitu *data* dan *mining*. Namun demikian, arti harfiah dari penggabungan dua kata tersebut tidak menjadi penambangan data. Pieter Adriaans and Dolf Zantinge memberikan definisinya mengenai *data mining* sebagai berikut [JEF04].

*“Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve their performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction.”*

Margaret H Dunham memaparkan deskripsinya mengenai *data mining* berikut ini [MAR03].

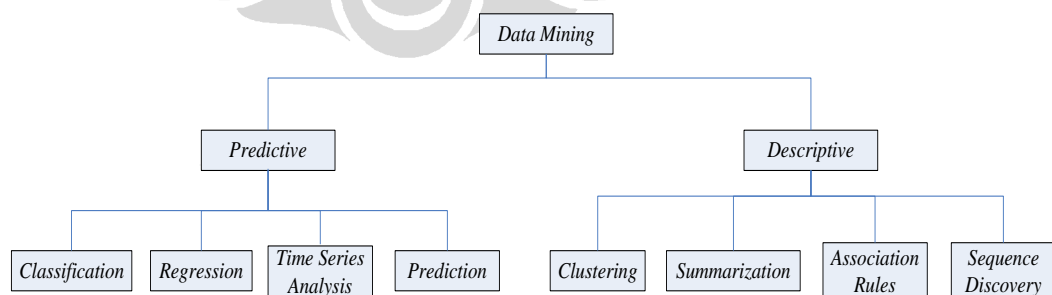
“Data mining is often defined as finding hidden information in a database. Alternatively, it has been called exploratory data analysis, data driven discovery, and deductive learning.”

Berdasarkan kedua definisi tersebut, *data mining* dapat diartikan sebagai suatu usaha untuk mengeksplorasi makna yang tersirat dalam suatu data melalui suatu metodologi tertentu.

Model yang dibuat dalam suatu proses *data mining* dapat bersifat prediktif maupun deskriptif. Model prediktif membuat sebuah prediksi mengenai nilai data dengan menggunakan hasil yang diketahui berdasarkan dari data yang berbeda-beda. Pemodelan prediktif dibuat berdasarkan penggunaan data-data yang telah ada sebelumnya. Misalnya dalam penolakan penggunaan suatu kartu kredit. Penolakan tersebut bukan disebabkan catatan pengguna kartu kredit tersebut, melainkan karena transaksi pembelian yang dilakukan saat ini sama dengan transaksi dari suatu kartu kredit curian. *Data mining* dengan model prediktif meliputi *classification*, *regression*, *time series analysis*, dan *prediction*.

Adapun model deskriptif mengidentifikasikan pola atau relasi dalam suatu data. Tidak seperti model prediktif, model ini menyediakan cara untuk mengeksplorasi data yang diujikan, bukan untuk memprediksi properti baru. Jenis-jenis algoritma yang digunakan adalah *clustering*, *summarization*, *association rules* dan *sequence discovery*.

Gambar 2.1 adalah ilustrasi dari deskripsi mengenai model prediktif dan deskriptif.



**Gambar 2.1: Metode Data Mining**

## 2.2 Metode *Data Mining*

Pemaparan tentang metode *data mining* dalam subbab ini merupakan sebagian dari berbagai macam metode *data mining*. Penjelasan ini terbatas pada metode yang digunakan oleh penulis dalam mengembangkan algoritma ini. Ada tiga metode yang akan dijelaskan yaitu *association rules*, *classification*, dan *clustering*.

### 2.2.1 *Association Rules*

*Association rules* dipopulerkan oleh Rakesh Agrawal, seorang peneliti di IBM sejak tahun 1993 [BOW06]. Metode ini sering digunakan oleh toko-toko retail untuk membantu pemasaran, penempatan, dan pengawasan produk serta pemasangan iklan. Meskipun dapat diaplikasikan dalam bisnis retail, metode ini dapat digunakan untuk kegunaan yang lain seperti memprediksi kesalahan dalam suatu jaringan telekomunikasi. Teknik ini digunakan untuk menampilkan relasi antara item data. Metode ini berusaha menemukan aturan-aturan tertentu yang mengasosiasikan antara suatu data dengan data yang lain.

Berikut ini adalah pendefinisian secara matematis dari beberapa terminologi dalam *association rules*.

- a. Misalkan ada sekumpulan item  $I = \{I_1, I_2, I_3, \dots, I_m\}$  dan suatu basis data transaksi  $D = \{t_1, t_2, t_3, \dots, t_n\}$  dimana  $t_i = \{I_{i1}, I_{i2}, I_{i3}, \dots, I_{ik}\}$  dan  $I_{ij} \in I$ , *association rules* adalah sebuah implikasi dari  $X \rightarrow Y$  dimana  $X$  dan  $Y$  merupakan himpunan bagian dari  $I$  adalah sekumpulan item yang dinamakan *itemset*.
- b. **Support** ( $s$ ) untuk suatu *association rules*  $X \rightarrow Y$  adalah persentase transaksi dalam suatu basis data yang memuat keseluruhan  $X$  dan  $Y$ . *Support* ini menunjukkan intensitas kemunculan suatu data dalam suatu *dataset*.
- c. **Confidence** atau **strength** ( $\alpha$ ) dalam suatu *association rules*  $X \rightarrow Y$  adalah perbandingan antara banyaknya transaksi yang memuat  $X$  dan  $Y$  dengan banyaknya transaksi yang memuat  $X$ .

Permasalahan dalam *association rules* ini adalah mengidentifikasi semua *association rules*  $X \rightarrow Y$  dengan suatu nilai *minimum support* dan *confidence*. Kedua nilai tersebut ( $s, \alpha$ ) adalah input yang diberikan. Efisiensi dari metode ini dihitung berdasarkan banyaknya proses *scan* yang dilakukan terhadap basis data dan nilai maksimum dari *itemset* yang harus dihitung.

Terdapat 3 algoritma dari metode *association* yang diimplementasikan ke dalam FIKUI Mining. Algoritma tersebut adalah algoritma *Apriori*, *CT-Pro*, dan *FP-Growth*. Ketiga algoritma ini akan dijelaskan lebih lanjut dalam bab 3 mengenai analisis dan perancangan sistem.

### 2.2.2 Classification

Metode ini adalah metode yang paling familiar dan populer dalam *data mining*. Contoh penerapan *classification* adalah pada pengenalan pola dan *image*, pendiagnosaan medis, pendeteksian kesalahan dalam dunia industri, dan pengklasifikasian tren pasar keuangan [BOW06].

*Classification* adalah proses untuk menemukan model atau fungsi yang menjelaskan atau membedakan konsep atau kelas data. Tujuannya untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Model itu sendiri dapat berupa aturan *if else*, *decision tree*, formula matematika, atau *neural network* [DAN07].

Berikut ini adalah pendefinisian secara matematis dari terminologi *classification*.

Dalam suatu basis data  $D = \{t_1, t_2, t_3, \dots, t_n\}$  dari suatu *tuples (items, records)* dan sebuah himpunan kelas-kelas  $C = \{C_1, C_2, C_3, \dots, C_m\}$ , permasalahan *classification* adalah untuk mendefinisikan suatu  $f: D \rightarrow C$  dimana untuk setiap  $t_i$  ditempatkan dalam satu kelas. Sebuah kelas,  $C_j$ , mengandung tepat *tuples* yang dipetakan kepadanya;  $C_j = \{t_i \mid f(t_i) = C_j, 1 \leq i \leq n, \text{ dan } t_i \in D\}$

Terdapat tiga metode dasar yang digunakan untuk menyelesaikan suatu permasalahan *classification*.

**a. Specifying boundaries**

*Classification* ini bekerja dengan membagi input dari *tuples* basis data potensial ke dalam area-area dimana tiap area diasosiasikan ke dalam satu kelas.

**b. Probability distributions**

Untuk setiap kelas yang diberikan,  $C_j$ ,  $P(t_i | C_j)$  adalah PDF untuk kelas yang dievaluasi di satu poin,  $t_i$ . Jika probabilitas peristiwa untuk setiap kelas  $P(C_j)$  diketahui, maka  $P(C_j) P(t_i | C_j)$  digunakan untuk memperkirakan probabilitas bahwa  $t_i$  ada di kelas  $C_j$ .

**c. Posterior probabilities**

Jika ada suatu nilai data  $t_i$ , kita dapat menemukan probabilitas bahwa  $t_i$  ada di kelas  $C_j$ . Ini ditunjukkan oleh  $P(C_j | t_i)$  dan disebut probabilitas posterior. Salah satu pendekatan *classification* akan menentukan probabilitas posterior untuk setiap kelas kemudian meng-assign  $t_i$  ke kelas yang memiliki probabilitas paling tinggi.

Ada tiga algoritma dari metode *classification* yang kami implementasikan ke dalam FIKUI Mining, yaitu *CMAR* dan *CSFP*. Kedua algoritma ini akan dijelaskan lebih lanjut dalam bab 3 mengenai analisis dan perancangan sistem.

### 2.2.3 Clustering

*Clustering* hamper sama dengan *classification* dimana data juga dikelompokkan. Namun, metode ini tidak seperti *classification*, pengelompokan yang dilakukan tidak didefinisikan sebelumnya. Pengelompokan dilakukan dengan mencari kesamaan antara data yang sesuai dengan karakteristik yang ditemukan dalam data sebenarnya. Kelompok-kelompok inilah yang dinamakan *clusters* [MAR03].

Berikut ini adalah pendefinisian secara matematis dari terminologi *clustering*.

Misalkan suatu basis data  $D = \{t_1, t_2, t_3, \dots, t_n\}$  dari *tuples* dan sebuah nilai integer  $k$ , permasalahan *clustering* adalah untuk mendefinisikan suatu

pemetaan  $f: D \rightarrow \{1, \dots, k\}$  dimana tiap  $t_i$  dipasangkan dengan satu *cluster*  $K_j$ ,  $1 \leq j \leq k$ . Suatu *cluster*,  $K_j$ , mengandung tepat *tuples* yang dipetakan; yaitu  $K_j = \{t_i \mid f(t_i) = K_j, 1 \leq i \leq n, \text{ dan } t_i \in D\}$

Terdapat 3 algoritma dari metode *clustering* yang diimplementasikan ke dalam FIKUI Mining, yaitu *Fuzzy C-Means*, *K-Means*, dan *Nearest Neighbour*. Ketiga algoritma tersebut akan dijelaskan lebih lanjut dalam bab 3 mengenai analisis dan perancangan sistem.



### 2.3 Use Case

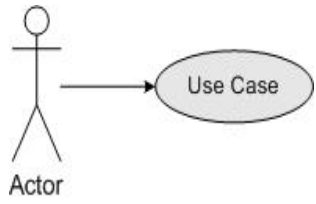
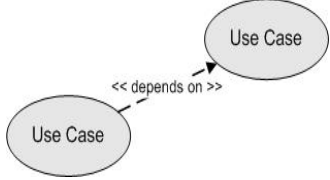
*Use case* adalah sebuah skenario atau peristiwa bisnis dimana sistem harus memberikan suatu respon yang ditentukan. *Use case* mencakup analisis berorientasi objek. Namun, penggunaannya menjadi hal umum dalam banyak metodologi untuk analisis dan perancangan sistem [WHI06].

*Use case diagram* adalah sebuah diagram yang menggambarkan interaksi dan keterhubungan antara sistem yang akan dibuat dengan sistem-sistem eksternal lainnya, termasuk *user* yang akan menggunakan sistem tersebut [WHI06].

Tabel 2.1 berisi notasi *use case diagram* beserta dengan penjelasan singkat dari notasi-notasi yang digunakan.

Tabel 2.1: Notasi *Use Case Diagram*

Notasi	Nama	Deskripsi
	<i>Use-Case</i>	Lambang dari suatu fungsi pada sistem. <i>Use-case</i> terdiri dari skenario yang dapat dilakukan oleh pengguna terhadap sistem untuk mencapai tujuan dari suatu proses bisnis.
	<i>Actor</i>	Pengguna yang menginisiasi terjadinya skenario pada <i>use-case</i> untuk menjalankan suatu proses bisnis pada sistem.

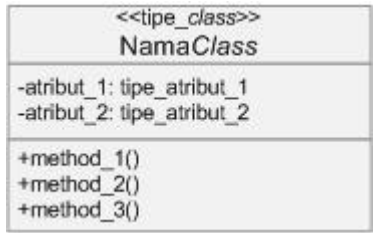

	<p style="text-align: center;"><i>Association</i></p>	<p>Notasi yang melambangkan suatu bentuk interaksi dan relasi antara <i>Actor</i> dan <i>Use Case</i>, dimana <i>Actor</i> menggunakan atau memicu berjalannya suatu fungsi di dalam <i>use-case</i> tersebut.</p>
	<p style="text-align: center;"><i>Dependency Relation</i></p>	<p>Notasi yang melambangkan relasi ketergantungan antar <i>Use Case</i>, dimana <i>Use Case</i> yang menunjuk, bergantung pada <i>Use Case</i> yang ditunjuk.</p>

## 2.4 Class Diagram

*Class diagram* adalah gambar mengenai struktur objek statis dari suatu sistem, menunjukkan kelas-kelas objek yang tersusun dalam sebuah sistem dan juga hubungan antara kelas objek tersebut [WHI06].

Tabel 2.2. adalah daftar notasi yang digunakan dalam merancang *Class Diagram*.

**Tabel 2.2: Notasi Class Diagram**

Notasi	Nama	Deskripsi
	<p style="text-align: center;"><i>Class</i></p>	<p>Kumpulan dari <i>instance</i> sebuah objek yang memiliki properti dan perilaku yang sama. Properti yang dimiliki dilambangkan dengan <i>attribute</i> dan perilaku atau operasi yang dapat dijalankan dilambangkan dengan <i>method</i>.</p>
	<p style="text-align: center;"><i>Message</i></p>	<p>Menunjukkan adanya komunikasi antar <i>class</i> berupa pemanggilan <i>method</i> dari satu <i>class</i> ke <i>class</i> lain.</p>
	<p style="text-align: center;"><i>Interface</i></p>	<p>Tampilan</p>

*Class* yang dirancang dibagi atas dua bagian yaitu *class* bertipe *controller* dan *model*. *Controller* merepresentasikan fitur-fitur yang menjadi fungsional

sistem, sedangkan *model* merepresentasikan setiap *entity* atau tabel pada basis data. Tidak seperti kelas *model* pada umumnya, atribut dari tabel tidak dijadikan sebagai properti dari kelas *model* yang bersesuaian. Hal ini disebabkan oleh kemampuan dari *framework* yang digunakan untuk membaca *meta-data* dari setiap tabel pada basis data. Kelas *model* cukup menyimpan hubungan *relationship* yang dimiliki suatu *entity* dengan *entity* lain.

Pemecahan *class diagram* untuk *model* ini dibagi menjadi dua bagian untuk meningkatkan kemudahan membaca *class diagram*. Pemecahan *class diagram* didasarkan pada pembagian *controller*. Pada bagian pertama, akan dijabarkan mengenai tiga *controller* beserta *model* yang diakses. Adapun pada bagian kedua, akan dijabarkan mengenai lima *controller* lain beserta *model-model* yang diakses oleh masing-masing *controller* tersebut. Karena adanya kesamaan *model* yang diakses oleh *controller* pada bagian pertama dengan bagian kedua, maka terdapat duplikasi dari *model* yang digambarkan pada kedua bagian diagram.

